UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/709,522	05/11/2004	Krishna Mohan ITIKARLAPALLI	ORCL-003	3521

51121 7590 04/29/2009

LAW FIRM OF NAREN THAPPETA
C/o Landon-IP Inc.,
1700 Diagonal Road, Suite 450
Alexandria, VA 22314

EXAMINER	
SANDERS, AARON J	

ART UNIT	PAPER NUMBER
2168	

NOTIFICATION DATE	DELIVERY MODE
04/29/2009	ELECTRONIC

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

Notice of the Office communication was sent electronically on above-indicated "Notification Date" to the following e-mail address(es):

lfnt2000@yahoo.com
oracle@iphorizons.com
intercomm@iphorizons.com

Office Action Summary	Application No.	Applicant(s)	
	10/709,522	ITIKARLAPALLI ET AL.	
	Examiner	Art Unit	
	AARON SANDERS	2168	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE *3* MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *11 February 2009*.

2a)☒ This action is **FINAL**. 2b)☐ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1-10,13-21 and 25* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1-10,13-21 and 25* is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☐ The drawing(s) filed on _____ is/are: a)☐ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☒ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☐ Information Disclosure Statement(s) (PTO/SB/08) Paper No(s)/Mail Date _____.
4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____ .
5) ☐ Notice of Informal Patent Application
6) ☐ Other: _____.

DETAILED ACTION

Response to Amendment

The amendment filed 11 February 2009 has been entered. Claims 1-10, 13-21, and 25 are pending. Claims 1, 5, 7-10, 13-16, 18, 20-21, and 25 are currently amended. Claims 11-12 and 22-24 are cancelled. No claims are new. This action is FINAL, as necessitated by amendment.

Claim Rejections - 35 USC § 112, First Paragraph

The following is a quotation of the first paragraph of 35 U.S.C. 112:

> The specification shall contain a written description of the invention, and of the manner and process of making and using it, in such full, clear, concise, and exact terms as to enable any person skilled in the art to which it pertains, or with which it is most nearly connected, to make and use the same and shall set forth the best mode contemplated by the inventor of carrying out his invention.

Claims 7 and 10 are rejected under 35 U.S.C. 112, first paragraph, as failing to comply with the written description requirement. The claim(s) contains subject matter which was not described in the specification in such a way as to reasonably convey to one skilled in the relevant art that the inventor(s), at the time the application was filed, had possession of the claimed invention. Specifically, the limitations "provides said identifier to said user program" and "providing said identifier to said user program" are new matter.

Claim Rejections - 35 USC § 103

The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and

> the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

Claims 1-2, 5-10, 13, 16-17, 20-21 and 25 are rejected under 35 U.S.C. 103(a) as being unpatentable over Applicant's Admitted Prior Art, Fig. 1 and Specification pars. 3-7 and 22-33 ("AAPA"), in view of Gostanian et al., U.S. 5,781,910 ("Gostanian"), and in view of Lordi et al., U.S. 5,857,204 ("Lordi").

1. AAPA teaches "*A method of implementing atomic transactions in a system, said method comprising*," see Fig. 1 and par. 23, "FIG. 1 contains pseudo-code illustrating the manner in which an example atomic transaction is implemented in a prior approach."

AAPA teaches "*specifying in said user program a plurality of combinations, wherein each of said plurality of combinations contains said transaction identifier, a task procedure, and a rollback procedure, wherein said task procedure implements a part of said atomic transaction and said rollback procedure is designed to rollback said task procedure*," see Fig. 1, par. 23, "For ease of understanding, atomic transaction Account1( ) (starting at line 105) is shown containing only few task procedures and desired roll-back procedures… Account1( ) is shown containing program logic in lines 110 through 199," par. 24, "Line 110 is shown containing a call to task procedure P1( ). Line 115 is shown containing a call to task procedure P2( )," and par. 25, "Lines 125 (do-reverse-of-P2( )) and 130 (do-reverse-of-P1( )) respectively represent roll-back procedures corresponding to P2( ) and P1( )," where the claimed "combinations" are the referenced Account( ), P( ), and do-reverse-of-P( ) combinations, and where Account( ) is the atomic transaction identifier. According to Applicant's specification at par. 69, "Even though the example

above are shown specifying the combination in the form of a single line of code (procedure call), multiple lines can be used in alternative embodiments." Thus, it is irrelevant that the referenced combinations are not contained in a single procedure call.

AAPA teaches "*executing a set of task procedures in a sequential order according to said user program, wherein said set of task procedures are contained in said task procedures specified in said plurality of combinations*," see Fig. 1 and par. 24, "Line 110 is shown containing a call to task procedure P1( ). Line 115 is shown containing a call to task procedure P2( ) and the status returned by execution of P2( ) is assigned to a variable Status."

AAPA teaches "*keeping track of a set of rollback procedures corresponding to said set of task procedures, each of said set of rollback procedures being determined based on a combination corresponding to an executed task procedure contained in said set of task procedures, said combination being contained in said plurality of combinations specified in said user program*," see Fig. 1, par. 7, "Thus, in one prior approach, a programmer may have to design programs to keep track of the specific tasks that have completed, and rollback the completed tasks if an atomic transaction is to be aborted," and par. 25, "Control passes to line 125 if an error has occurred, to line 140 otherwise. Lines 125 (do-reverse-of-P2( )) and 130 (do-reverse-of-P1( )) respectively represent roll-back procedures corresponding to P2( ) and P1( )," where the claimed "combination" is, for example, the referenced Account( ), P1( ), and do-reverse-of-P1( ) combination. AAPA does not teach "*wherein said set of rollback procedures are kept track of external to said user program in response to said executing of the corresponding task procedures*." Lordi does, however, see Fig. 2 and col. 5, ll. 50-62, "A Perform

routine 100 for an operation takes the same parameters as does the corresponding native routine and generally executes the following steps… makes a log entry by creating the entry and appending it to a transaction log (a log database), the entry containing information needed to commit and roll back, including the information required to call the Finalize and Undo routines," where the claimed "task procedure" is the referenced "Perform routine" and the claimed "rollback procedure" is the referenced "Undo routine." The referenced log "keeps track" of the Undo routines and is external to the user program, as it is a "log database." Thus, it would have been obvious to one of ordinary skill in the database art at the time of the invention to combine the teachings of the cited references because Lordi's teachings would have allowed AAPA's method to gain permanent, non-volatile storage of the information and procedures required to rollback the system to a previous state without relying on the possibly crashed user program, as in AAPA, see Lordi col. 6, ll. 3-11.

AAPA teaches "*and executing said set of rollback procedures in a reverse order of said sequential order if said atomic transaction is to be aborted*," see Fig. 1 and par. 24, "Control passes to line 125 if an error has occurred, to line 140 otherwise. Lines 125 (do-reverse-of-P2( )) and 130 (do-reverse-of-P1( )) respectively represent roll-back procedures corresponding to P2( ) and P1( )."

AAPA teaches "*wherein said rollback procedure is specified as a separate procedure from said task procedure in said user program*," see Fig. 1 and par. 25, "Lines 125 (do-reverse-of-P2( )) and 130 (do-reverse-of-P1( )) respectively represent roll-back procedures corresponding to P2( ) and P1( )."

AAPA teaches "*wherein said user program contains groups of instructions to implement respective program logic for each of said task procedure and said rollback procedure*," see Fig. 1 and par. 23, "For ease of understanding, atomic transaction Account1( ) (starting at line 105) is shown containing only few task procedures and desired roll-back procedures. However, typical atomic transactions contain many task procedures," where the claimed "groups of instructions" are contained in the referenced "procedures," see Applicant's specification par. 35, which defines a procedure as "a group of instructions identified by a name."

AAPA teaches "*and whereby each user program has corresponding custom logic specified by a user for each of the rollback procedure*," see Fig. 1 and par. 23, "FIG. 1 contains pseudo-code illustrating the manner in which an example atomic transaction is implemented in a prior approach." Since a user writes the code, that user could add custom logic to the procedures.

AAPA does not teach "*requesting in a user program a transaction identifier for an atomic transaction*." Gostanian does, however, see Figs. 3, 5, col. 9, lines 1-21, "Each application client 302-308 is essentially an application program that preferably resides on a client computer 220 (FIG. 2)," col. 9, lines 27-42, "The application servers 332, 334 coordinate the requested database transactions for the application clients 302-308" and col. 13, line 61 – col. 14, line 9, "As with the 1PPC protocol 400 (FIG. 4), a manager process 516 of the coordinator 512 first assigns a unique transaction identification code 524 to the particular transaction," where the claimed "user program" is the referenced "application program." Thus, it would have been obvious to one of ordinary skill in the database art at the time of the invention to combine the teachings of the cited references

because Gostanian's teachings would have allowed AAPA's method to gain a common means of identifying transactions, see Lordi col. 13, ll. 48-62.

AAPA does not teach "*generating said transaction identifier in a transaction manager in response to said requesting*." Gostanian does, however, see Fig. 5 and col. 13, line 61 – col. 14, line 9, "As with the 1PPC protocol 400 (FIG. 4), a manager process 516 of the coordinator 512 first assigns a unique transaction identification code 524 to the particular transaction." Thus, it would have been obvious to one of ordinary skill in the database art at the time of the invention to combine the teachings of the cited references because Gostanian's teachings would have allowed AAPA's method to gain a common means of identifying transactions, see Lordi col. 13, ll. 48-62.

2. AAPA does not teach "*The method of claim 1, wherein said transaction identifier is unique to each of the atomic transactions*." Gostanian does, however, see Fig. 5 and col. 13, line 61 – col. 14, line 9, "As with the 1PPC protocol 400 (FIG. 4), a manager process 516 of the coordinator 512 first assigns a unique transaction identification code 524 to the particular transaction." Thus, it would have been obvious to one of ordinary skill in the database art at the time of the invention to combine the teachings of the cited references because Gostanian's teachings would have allowed AAPA's method to gain a common means of identifying transactions, see Lordi col. 13, ll. 48-62.

5. AAPA teaches "*The method of claim 1, wherein said user program further comprises additional instruction for examining a status returned by execution of one of said task procedures and performing said aborting if said status indicates an error*," see Fig. 1 and par. 25, "In line 120, the variable Status is compared with ERROR1 (either a

variable set ahead, or a constant value defined elsewhere) to determine whether an error has occurred in the execution of P2( )."

AAPA does not teach "*wherein said aborting is specified in said user program using an instruction containing said transaction identifier*." Lordi does, however, see col. 2, l. 66 - col. 3, l. 7, "To roll a transaction back (Step 46), the transaction master broadcasts a roll back message (including the identifier for the transaction) to all agents that participated in the transaction (Step 48). Each agent steps through its log (Step 50), and for each operation belonging to the transaction being rolled back, an undo routine may be invoked which will have the effect of undoing the effects of the original operation (Step 52)." Thus, it would have been obvious to one of ordinary skill in the database art at the time of the invention to combine the teachings of the cited references because Lordi's teachings would have allowed AAPA's method to gain permanent, non-volatile storage of the information and procedures required to rollback the system to a previous state without relying on the possibly crashed user program, as in AAPA, see Lordi col. 6, ll. 3-11.

6. AAPA teaches "*The method of claim 1, wherein said aborting is performed asynchronously*," see Fig. 1 and par. 28, "In line 160, the variable Status is compared with ERROR2 (which is similar to ERROR1, described above) to determine whether an error has occurred in the execution of P6( )."

7. AAPA teaches "*A computer readable storage medium carrying one or more sequences of instructions representing a user program for execution on a system, said user program implementing an atomic transaction, wherein execution of said one or more sequences of instructions by one or more processors contained in said system*

causes said system to perform the actions of," see Fig. 1 and par. 23, "FIG. 1 contains pseudo-code illustrating the manner in which an example atomic transaction is implemented in a prior approach."

AAPA teaches "*specifying a plurality of combinations in said user program for execution in said system, wherein each of said plurality of combinations contains said variable, a task procedure, and a rollback procedure, wherein said task procedure implements a part of said atomic transaction and said rollback procedure is designed to rollback said task procedure, wherein said variable in each of said plurality of combinations specifies said identifier generated by said transaction manager*," see Fig. 1, par. 23, "For ease of understanding, atomic transaction Account1( ) (starting at line 105) is shown containing only few task procedures and desired roll-back procedures… Account1( ) is shown containing program logic in lines 110 through 199," par. 24, "Line 110 is shown containing a call to task procedure P1( ). Line 115 is shown containing a call to task procedure P2( )," and par. 25, "Lines 125 (do-reverse-of-P2( )) and 130 (do-reverse-of-P1( )) respectively represent roll-back procedures corresponding to P2( ) and P1( )," where the claimed "combinations" are the referenced Account( ), P( ), and do-reverse-of-P( ) combinations, and where Account( ) is the atomic transaction identifier. According to Applicant's specification at par. 69, "Even though the example above are shown specifying the combination in the form of a single line of code (procedure call), multiple lines can be used in alternative embodiments." Thus, it is irrelevant that the referenced combinations are not contained in a single procedure call.

AAPA teaches "*wherein said plurality of combinations and said abort procedure are contained in a said user program*," see Fig. 1 and par. 25, "Lines 125 (do-reverse-of-

P2( )) and 130 (do-reverse-of-P1( )) respectively represent roll-back procedures corresponding to P2( ) and P1( ).”

AAPA teaches “*wherein said user program contains groups of instructions to implement respective program logic for each of said task procedure and said rollback procedure*.” AAPA does, however, see Fig. 1 and par. 23, “For ease of understanding, atomic transaction Account1( ) (starting at line 105) is shown containing only few task procedures and desired roll-back procedures. However, typical atomic transactions contain many task procedures,” where the claimed “groups of instructions” are contained in the referenced “procedures,” see Applicant’s specification par. 35, which defines a procedure as “a group of instructions identified by a name.”

AAPA teaches “*whereby each user program has corresponding custom logic specified by a user for each of the rollback procedure*,” see Fig. 1 and par. 23, “FIG. 1 contains pseudo-code illustrating the manner in which an example atomic transaction is implemented in a prior approach.” Since a programmer writes the code, that programmer could add custom logic to the procedures.

AAPA does not teach “*requesting an identifier in said user program from a transaction manager for said atomic transaction, wherein said transaction manager generates a unique value as said identifier and provides said identifier to said user program*.” Gostanian does, however, see Figs. 3, 5, col. 9, lines 1-21, “Each application client 302-308 is essentially an application program that preferably resides on a client computer 220 (FIG. 2),” col. 9, lines 27-42, “The application servers 332, 334 coordinate the requested database transactions for the application clients 302-308” and col. 13, line 61 – col. 14, line 9, “As with the 1PPC protocol 400 (FIG. 4), a manager process 516 of

the coordinator 512 first assigns a unique transaction identification code 524 to the particular transaction," where the claimed "user program" is the referenced "application program."

AAPA does not teach "*setting a variable to equal said identifier in said user program.*" Lordi does, however, see "In a second way of deploying the routines, a runtime library is created containing the routines, plus routines to start, finalize, and undo transactions… Upon abnormal termination, the application invokes the roll back process, which in turn scans the log and invokes the appropriate Undo routines," where the claimed "user program" is the claimed "routines" and it would have been obvious to to "set a variable to equal said identifier" so that the Perform and Undo routines would know which transactions to perform or undo. Thus, it would have been obvious to one of ordinary skill in the database art at the time of the invention to combine the teachings of the cited references because Lordi's teachings would have allowed AAPA's method to gain permanent, non-volatile storage of the information and procedures required to rollback the system to a previous state without relying on the possibly crashed user program, as in AAPA, see Lordi col. 6, ll. 3-11.

AAPA does not teach "*and aborting said atomic transaction by specifying, in said user program, said identifier associated with an abort procedure to cause said rollback procedures to be executed.*" Lordi does, however, see col. 2, l. 66 - col. 3, l. 7, "To roll a transaction back (Step 46), the transaction master broadcasts a roll back message (including the identifier for the transaction) to all agents that participated in the transaction (Step 48). Each agent steps through its log (Step 50), and for each operation belonging to the transaction being rolled back, an undo routine may be invoked which

will have the effect of undoing the effects of the original operation (Step 52)." Thus, it would have been obvious to one of ordinary skill in the database art at the time of the invention to combine the teachings of the cited references because Lordi's teachings would have allowed AAPA's method to gain permanent, non-volatile storage of the information and procedures required to rollback the system to a previous state without relying on the possibly crashed user program, as in AAPA, see Lordi col. 6, ll. 3-11.

8. AAPA teaches "*The computer readable storage medium of claim 7, wherein said specifying comprises including each of said plurality of combinations in a single procedure call*," see Fig. 1 and par. 23, "For ease of understanding, atomic transaction Account1( ) (starting at line 105) is shown containing only few task procedures and desired roll-back procedures… Account1( ) is shown containing program logic in lines 110 through 199," where the claimed "single procedure call" is the referenced "Account1( )."

9. AAPA teaches "*The computer readable storage medium of claim 7, further comprising examining a status returned by execution of one of said task procedures and performing said aborting if said status indicates an error*," see Fig. 1 and par. 25, "In line 120, the variable Status is compared with ERROR1 (either a variable set ahead, or a constant value defined elsewhere) to determine whether an error has occurred in the execution of P2( )."

10. AAPA teaches "*A computer readable storage medium carrying one or more sequences of instructions for supporting implementation of an atomic transaction in a system, wherein execution of said one or more sequences of instructions by one or more processors contained in said system causes said system to perform the actions of*," see

Fig. 1 and par. 23, "FIG. 1 contains pseudo-code illustrating the manner in which an example atomic transaction is implemented in a prior approach."

AAPA teaches "*receiving a plurality of combinations for execution from said user program, wherein each of said plurality of combinations contains said transaction identifier, a task procedure, and a rollback procedure, wherein said task procedure implements a part of said atomic transaction and said rollback procedure is designed to rollback said task procedure*," see Fig. 1, par. 23, "For ease of understanding, atomic transaction Account1( ) (starting at line 105) is shown containing only few task procedures and desired roll-back procedures… Account1( ) is shown containing program logic in lines 110 through 199," par. 24, "Line 110 is shown containing a call to task procedure P1( ). Line 115 is shown containing a call to task procedure P2( )," and par. 25, "Lines 125 (do-reverse-of-P2( )) and 130 (do-reverse-of-P1( )) respectively represent roll-back procedures corresponding to P2( ) and P1( )," where the claimed "combinations" are the referenced Account( ), P( ), and do-reverse-of-P( ) combinations, and where Account( ) is the atomic transaction identifier. According to Applicant's specification at par. 69, "Even though the example above are shown specifying the combination in the form of a single line of code (procedure call), multiple lines can be used in alternative embodiments." Thus, it is irrelevant that the referenced combinations are not contained in a single procedure call.

AAPA teaches "*executing a set of task procedures in a sequential order according to said user program, wherein said set of task procedures are contained in said plurality of combinations*," see Fig. 1 and par. 24, "Line 110 is shown containing a

call to task procedure P1( ). Line 115 is shown containing a call to task procedure P2( ) and the status returned by execution of P2( ) is assigned to a variable Status."

AAPA teaches "*keeping track of a set of rollback procedures corresponding to said set of task procedures, each of said set of rollback procedures being determined based on a combination corresponding to an executed task procedure contained in said set of task procedures, said combination being contained in said plurality of combinations specified in said user program*," see Fig. 1 and par. 25, "Control passes to line 125 if an error has occurred, to line 140 otherwise. Lines 125 (do-reverse-of-P2( )) and 130 (do-reverse-of-P1( )) respectively represent roll-back procedures corresponding to P2( ) and P1( )," where the claimed "combination" is, for example, the referenced Account( ), P1( ), and do-reverse-of-P1( ) combination, and where Account( ) is the atomic transaction identifier.

AAPA teaches "*and executing said set of rollback procedures in a reverse order of said sequential order in response to receiving an abort request*," see Fig. 1 and par. 24, "Control passes to line 125 if an error has occurred, to line 140 otherwise. Lines 125 (do-reverse-of-P2( )) and 130 (do-reverse-of-P1( )) respectively represent roll-back procedures corresponding to P2( ) and P1( )." AAPA does not teach "*said abort request being received from said user program and containing said identifier*." Lordi does, however, see col. 2, l. 66 - col. 3, l. 7, "To roll a transaction back (Step 46), the transaction master broadcasts a roll back message (including the identifier for the transaction) to all agents that participated in the transaction (Step 48). Each agent steps through its log (Step 50), and for each operation belonging to the transaction being rolled back, an undo routine may be invoked which will have the effect of undoing the effects

of the original operation (Step 52)." Thus, it would have been obvious to one of ordinary skill in the database art at the time of the invention to combine the teachings of the cited references because Lordi's teachings would have allowed AAPA's method to gain permanent, non-volatile storage of the information and procedures required to rollback the system to a previous state without relying on the possibly crashed user program, as in AAPA, see Lordi col. 6, ll. 3-11.

AAPA teaches "*wherein said rollback procedure is specified as a separate procedure from said task procedure in said user program*," see Fig. 1 and par. 25, "Lines 125 (do-reverse-of-P2( )) and 130 (do-reverse-of-P1( )) respectively represent roll-back procedures corresponding to P2( ) and P1( )."

AAPA teaches "*wherein said user program contains groups of instructions to implement respective program logic for each of said task procedure and said rollback procedure*," see Fig. 1 and par. 23, "For ease of understanding, atomic transaction Account1( ) (starting at line 105) is shown containing only few task procedures and desired roll-back procedures. However, typical atomic transactions contain many task procedures," where the claimed "groups of instructions" are contained in the referenced "procedures," see Applicant's specification par. 35, which defines a procedure as "a group of instructions identified by a name."

AAPA teaches "*whereby each user program has corresponding custom logic specified by a user for each of the rollback procedures*," see Fig. 1 and par. 23, "FIG. 1 contains pseudo-code illustrating the manner in which an example atomic transaction is implemented in a prior approach." Since a programmer writes the code, that programmer could add custom logic to the procedures.

AAPA does not teach "*generating an identifier for said atomic transaction for a user program*." Gostanian does, however, see Fig. 5 and col. 13, line 61 – col. 14, line 9, "As with the 1PPC protocol 400 (FIG. 4), a manager process 516 of the coordinator 512 first assigns a unique transaction identification code 524 to the particular transaction." Thus, it would have been obvious to one of ordinary skill in the database art at the time of the invention to combine the teachings of the cited references because Gostanian's teachings would have allowed AAPA's method to gain a common means of identifying transactions, see Lordi col. 13, ll. 48-62.

AAPA does not teach "*providing said identifier to said user program*." Gostanian does, however, see Fig. 5 and col. 13, l. 61 – col. 14, l. 9, "As with the 1PPC protocol 400 (FIG. 4), a manager process 516 of the coordinator 512 first assigns a unique transaction identification code 524 to the particular transaction. Next, as shown by block 526, the manager process 516 forwards the transaction to each cohort 514 using an atomic multicast message 528." Thus, it would have been obvious to one of ordinary skill in the database art at the time of the invention to combine the teachings of the cited references because Gostanian's teachings would have allowed AAPA's method to gain a common means of identifying transactions, see Lordi col. 13, ll. 48-62.

13. AAPA does not teach "*The computer readable storage medium of claim 10, wherein said transaction identifier is generated to be unique for each atomic transaction*." Gostanian does, however, see Fig. 5 and col. 13, line 61 – col. 14, line 9, "As with the 1PPC protocol 400 (FIG. 4), a manager process 516 of the coordinator 512 first assigns a unique transaction identification code 524 to the particular transaction." Thus, it would have been obvious to one of ordinary skill in the database art at the time of

the invention to combine the teachings of the cited references because Gostanian's teachings would have allowed AAPA's method to gain a common means of identifying transactions, see Lordi col. 13, ll. 48-62.

16. AAPA teaches "*a computer readable medium to store and provide said plurality of instructions to said memory, wherein execution of said plurality of instructions by said processing unit causes said computer system to support implementation of atomic transactions in a programming environment by performing the operations of*," see Fig. 1 and par. 23, "FIG. 1 contains pseudo-code illustrating the manner in which an example atomic transaction is implemented in a prior approach."

AAPA teaches "*specify in said user program a plurality of combinations for execution in a sequential order, wherein each of said plurality of combinations contains said transaction identifier, a task procedure, and a rollback procedure, wherein said task procedure implements a part of said atomic transaction and said rollback procedure is designed to rollback said task procedure, wherein said rollback procedure is specified as a separate procedure from said task procedure*," see Fig. 1, par. 23, "For ease of understanding, atomic transaction Account1( ) (starting at line 105) is shown containing only few task procedures and desired roll-back procedures… Account1( ) is shown containing program logic in lines 110 through 199," par. 24, "Line 110 is shown containing a call to task procedure P1( ). Line 115 is shown containing a call to task procedure P2( )," and par. 25, "Lines 125 (do-reverse-of-P2( )) and 130 (do-reverse-of-P1( )) respectively represent roll-back procedures corresponding to P2( ) and P1( )," where the claimed "combinations" are the referenced Account( ), P( ), and do-reverse-of-P( ) combinations, and where Account( ) is the atomic transaction identifier. According

to Applicant's specification at par. 69, "Even though the example above are shown specifying the combination in the form of a single line of code (procedure call), multiple lines can be used in alternative embodiments." Thus, it is irrelevant that the referenced combinations are not contained in a single procedure call.

AAPA teaches "*execute a set of task procedures in a sequential order according to said user program*," see Fig. 1 and par. 24, "Line 110 is shown containing a call to task procedure P1( ). Line 115 is shown containing a call to task procedure P2( ) and the status returned by execution of P2( ) is assigned to a variable Status."

AAPA teaches "*keep track of a set of rollback procedures corresponding to said set of task procedures, each of said set of rollback procedures being determined based on a combination corresponding to an executed task procedure contained in said set of task procedures, said combination being contained in said plurality of combinations specified in said user program*," see Fig. 1 and par. 25, "Control passes to line 125 if an error has occurred, to line 140 otherwise. Lines 125 (do-reverse-of-P2( )) and 130 (do-reverse-of-P1( )) respectively represent roll-back procedures corresponding to P2( ) and P1( )," where the claimed "combination" is, for example, the referenced Account( ), P1( ), and do-reverse-of-P1( ) combination, and where Account( ) is the atomic transaction identifier. AAPA does not teach "*wherein said set of rollback procedures are kept track of external to said user program in response to said executing of the corresponding task procedures*." Lordi does, however, see Fig. 2 and col. 5, ll. 50-62, "A Perform routine 100 for an operation takes the same parameters as does the corresponding native routine and generally executes the following steps… makes a log entry by creating the entry and appending it to a transaction log (a log database), the entry containing information needed

to commit and roll back, including the information required to call the Finalize and Undo routines," where the claimed "task procedure" is the referenced "Perform routine" and the claimed "rollback procedure" is the referenced "Undo routine." The referenced log "keeps track" of the Undo routines and is external to the user program, as it is a "log database." Thus, it would have been obvious to one of ordinary skill in the database art at the time of the invention to combine the teachings of the cited references because Lordi's teachings would have allowed AAPA's method to gain permanent, non-volatile storage of the information and procedures required to rollback the system to a previous state without relying on the possibly crashed user program, as in AAPA, see Lordi col. 6, ll. 3-11.

AAPA teaches "*and execute said set of rollback procedures in a reverse order of said sequential order if said atomic transaction is to be aborted, wherein said rollback procedures are identified according to said keeping*," see Fig. 1 and par. 24, "Control passes to line 125 if an error has occurred, to line 140 otherwise. Lines 125 (do-reverse-of-P2( )) and 130 (do-reverse-of-P1( )) respectively represent roll-back procedures corresponding to P2( ) and P1( )."

AAPA teaches "*wherein said user program contains groups of instructions to implement respective program logic for each of said task procedure and said rollback procedure*," see Fig. 1 and par. 23, "For ease of understanding, atomic transaction Account1( ) (starting at line 105) is shown containing only few task procedures and desired roll-back procedures. However, typical atomic transactions contain many task procedures," where the claimed "groups of instructions" are contained in the referenced

"procedures," see Applicant's specification par. 35, which defines a procedure as "a group of instructions identified by a name."

AAPA teaches "*whereby each user program has corresponding custom logic specified by a user for each of the rollback procedures*," see Fig. 1 and par. 23, "FIG. 1 contains pseudo-code illustrating the manner in which an example atomic transaction is implemented in a prior approach." Since a programmer writes the code, that programmer could add custom logic to the procedures.

AAPA does not teach "*A computer system comprising*." Gostanian does, however, see Fig. 2 and col. 7, lines 46-62, "FIG. 2 is a block diagram of a database system 200." Thus, it would have been obvious to one of ordinary skill in the database art at the time of the invention to combine the teachings of the cited references because Gostanian's teachings would have allowed AAPA's method to gain a common means of implementing database operations, see Gostanian col. 7, ll. 46-62.

AAPA does not teach "*a memory storing a plurality of instructions*," see Fig. 2 and col. 8, lines 14-24, "As shown in FIG. 2, a typical hardware configuration of a client 220 includes a central processing unit (CPU) 222 coupled between a memory 224." Thus, it would have been obvious to one of ordinary skill in the database art at the time of the invention to combine the teachings of the cited references because Gostanian's teachings would have allowed AAPA's method to gain a common means of implementing database operations, see Gostanian col. 7, ll. 46-62.

AAPA does not teach "*and a processing unit coupled to said memory and executing said plurality of instructions*." Gostanian does, however, see Fig. 2, e.g. "CPU" 222. Thus, it would have been obvious to one of ordinary skill in the database art

at the time of the invention to combine the teachings of the cited references because Gostanian's teachings would have allowed AAPA's method to gain a common means of implementing database operations, see Gostanian col. 7, ll. 46-62.

AAPA does not teach "*request in a user program a transaction identifier for an atomic transaction*." Gostanian does, however, see Figs. 3, 5, col. 9, lines 1-21, "Each application client 302-308 is essentially an application program that preferably resides on a client computer 220 (FIG. 2)," col. 9, lines 27-42, "The application servers 332, 334 coordinate the requested database transactions for the application clients 302-308" and col. 13, line 61 – col. 14, line 9, "As with the 1PPC protocol 400 (FIG. 4), a manager process 516 of the coordinator 512 first assigns a unique transaction identification code 524 to the particular transaction," where the claimed "user program" is the referenced "application program." Thus, it would have been obvious to one of ordinary skill in the database art at the time of the invention to combine the teachings of the cited references because Gostanian's teachings would have allowed AAPA's method to gain a common means of identifying transactions, see Lordi col. 13, ll. 48-62.

AAPA does not teach "*generate said transaction identifier in a transaction manager in response to said requesting, wherein said transaction manager is provided external to said user program*." Gostanian does, however, see Fig. 5 and col. 13, line 61 – col. 14, line 9, "As with the 1PPC protocol 400 (FIG. 4), a manager process 516 of the coordinator 512 first assigns a unique transaction identification code 524 to the particular transaction." Thus, it would have been obvious to one of ordinary skill in the database art at the time of the invention to combine the teachings of the cited references because

Gostanian's teachings would have allowed AAPA's method to gain a common means of identifying transactions, see Lordi col. 13, ll. 48-62.

17. AAPA does not teach "*The computer system of claim 16, wherein said transaction identifier is unique to each of the atomic transactions*." Gostanian does, however, see Fig. 5 and col. 13, line 61 – col. 14, line 9, "As with the 1PPC protocol 400 (FIG. 4), a manager process 516 of the coordinator 512 first assigns a unique transaction identification code 524 to the particular transaction." Thus, it would have been obvious to one of ordinary skill in the database art at the time of the invention to combine the teachings of the cited references because Gostanian's teachings would have allowed AAPA's method to gain a common means of identifying transactions, see Lordi col. 13, ll. 48-62.

20. AAPA teaches "*The computer system of claim 16, wherein the actions performed by said computer system further comprise examine a status returned by execution of one of said task procedures and to perform said aborting if said status indicates an error*," see Fig. 1.

21. Gostanian teaches "*The computer system of claim 16, wherein the actions performed by said computer system further comprise execute said rollback procedures asynchronously*," see Fig. 1.

25. AAPA teaches "*The computer readable storage medium of claim 7, wherein said rollback procedure is specified as a separate procedure from said task procedure in said user program*," see Fig. 1.

Claims 3-4, 14-15 and 18-19 are rejected under 35 U.S.C. 103(a) as being unpatentable over Applicant's Admitted Prior Art, Fig. 1 and Specification pars. 3-7 and 22-33 ("AAPA"), in view of Gostanian et al., U.S. 5,781,910 ("Gostanian"), in view of Lordi et al., U.S. 5,857,204 ("Lordi"), and in view of Raz, U.S. 5,701,480 ("Raz").

3. AAPA does not teach "*The method of claim 1, wherein said keeping comprises storing data representing said rollback procedures in a stack.*" Raz does, however, see col. 19, lines 51-59, "the transaction scheduler responds to an interrupt by removing the context of the interrupted transaction from the processor stack of the digital computer… The context includes the value of the program counter which points to the interrupted memory location in the transaction program." Thus, it would have been obvious to one of ordinary skill in the database art at the time of the invention to combine the teachings of the cited references because Raz's teachings would have allowed AAPA's method to gain way to keep track of transactions, see Raz col. 19, lines 51-59.

4. AAPA does not teach "*The method of claim 3, wherein said stack is stored in a memory.*" Raz does, however, see col. 2, lines 7-24, "the operating system typically provides an established set of memory management procedures that can be invoked or called from an application program to define a 'recovery unit,'" where the "stack" in the reference is part of the "recovery unit." Thus, it would have been obvious to one of ordinary skill in the database art at the time of the invention to combine the teachings of the cited references because Raz's teachings would have allowed AAPA's method to gain way to keep track of transactions, see Raz col. 19, lines 51-59.

14. AAPA does not teach "*The computer readable storage medium of claim 10, wherein said set of rollback procedures are represented in the form of a stack.*" Raz

does, however, see col. 19, lines 51-59, "the transaction scheduler responds to an interrupt by removing the context of the interrupted transaction from the processor stack of the digital computer… The context includes the value of the program counter which points to the interrupted memory location in the transaction program." Thus, it would have been obvious to one of ordinary skill in the database art at the time of the invention to combine the teachings of the cited references because Raz's teachings would have allowed AAPA's method to gain way to keep track of transactions, see Raz col. 19, lines 51-59.

15. AAPA does not teach "T*he computer readable storage medium of claim 14, wherein said stack is stored in a memory*." Raz does, however, see col. 2, lines 7-24, "the operating system typically provides an established set of memory management procedures that can be invoked or called from an application program to define a 'recovery unit'", where the "stack" in the reference is part of the "recovery unit." Thus, it would have been obvious to one of ordinary skill in the database art at the time of the invention to combine the teachings of the cited references because Raz's teachings would have allowed AAPA's method to gain way to keep track of transactions, see Raz col. 19, lines 51-59.

18. AAPA does not teach "*The computer system of claim 16, wherein the actions performed by said computer system further comprise store data representing said rollback procedures in a stack to perform said keep*." Raz does, however, see col. 19, lines 51-59, "the transaction scheduler responds to an interrupt by removing the context of the interrupted transaction from the processor stack of the digital computer… The context includes the value of the program counter which points to the interrupted memory

location in the transaction program." Thus, it would have been obvious to one of ordinary skill in the database art at the time of the invention to combine the teachings of the cited references because Raz's teachings would have allowed AAPA's method to gain way to keep track of transactions, see Raz col. 19, lines 51-59.

19. AAPA do not teach "*The computer system of claim 18, wherein said stack is stored in a memory*." Raz does, however, see col. 2, lines 7-24, "the operating system typically provides an established set of memory management procedures that can be invoked or called from an application program to define a 'recovery unit,'" where the "stack" in the reference is part of the "recovery unit." Thus, it would have been obvious to one of ordinary skill in the database art at the time of the invention to combine the teachings of the cited references because Raz's teachings would have allowed AAPA's method to gain way to keep track of transactions, see Raz col. 19, lines 51-59.

Response to Arguments

As per Applicant's argument that AAPA's "Account()" in Fig. 1 cannot be a transaction identifier, the Examiner respectfully disagrees. First, Account() is described as an "atomic transaction… containing only few task procedures and desired roll-back procedures… [and] containing program logic in lines 110 through 199," see par. 23. Clearly then, Account() identifies the transaction. Further, each of the claimed "combinations" do not require their own unique identifier. There is nothing in the claims that precludes the interpretation that one combination is Account(), P1(), and do-reverse-of-P1(), Account(), P2(), and do-reverse-of-P2(), Account(), P3(), and do-reverse-of-P3(),

etc. Finally, when combined with Gostanian and Lordi, a transaction manager would generate the name Account() to describe the transaction.

Applicant's remaining arguments with respect to the U.S.C. 103 rejections of the independent claims have been considered but are moot in view of the new ground(s) of rejection.

Conclusion

The prior art made of record and not relied upon is considered pertinent to Applicant's disclosure: U.S. 2002/0007363.

Applicant's amendment necessitated the new ground(s) of rejection presented in this Office action. Accordingly, **THIS ACTION IS MADE FINAL**. See MPEP § 706.07(a). Applicant is reminded of the extension of time policy as set forth in 37 CFR 1.136(a).

A shortened statutory period for reply to this final action is set to expire THREE MONTHS from the mailing date of this action. In the event a first reply is filed within TWO MONTHS of the mailing date of this final action and the advisory action is not mailed until after the end of the THREE-MONTH shortened statutory period, then the shortened statutory period will expire on the date the advisory action is mailed, and any extension fee pursuant to 37 CFR 1.136(a) will be calculated from the mailing date of the advisory action. In no event, however, will the statutory period for reply expire later than SIX MONTHS from the date of this final action.

Any inquiry concerning this communication or earlier communications from the examiner should be directed to Aaron Sanders whose telephone number is 571-270-1016. The examiner can normally be reached on M-F 9:00a-4:00p.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Tim Vo can be reached on 571-272-3642. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free). If you would like assistance from a USPTO Customer Service Representative or access to the automated information system, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.

/Tim T. Vo/
Supervisory Patent Examiner, Art Unit 2168

/Aaron Sanders/
Examiner, Art Unit 2168
14 April 2009